XSEDE Federation and Interoperation Use Cases

22 January 2013

Version 0.3

Table of Contents

A Document History

Overall Document Authors:

Altaf Hossain
PSC
altaf@psc.edu

Shantenu Jha
Rutgers University
shantenu.jha@rutgers.edu

Ole Weidner
Rutgers University
ole.weidner@rutgers.edu

	Version	Date	Changes	Author
First use case draft	0.1	12/01/2012	Document created	Jha, Weidner
Draft update on feedback	0.2	12/04/2012	List of science use cases; glossary; draft development of first 2 use cases	Jha, Weidner, Hossain
Draft update on feedback	0.3	2/04/2013	Formatting, refinement of use cases	Weidner

B Document Scope

The use cases are presented here using the following format, derived from the Malan and Bredemeyer white paper[1] as follows:

Use Case	UCFI 1.0
Description	A loosely coupled / ensemble / workflow / etc. application requires (could hugely benefit from) more computational resources than one single machine can provide. The application user (could also be: portal, gateway) has allocations on multiple XSEDE resources, so he/she has a strong interest to use more than one resource concurrently, either programmatically or via simple cmd-line tools.
References	● Distributed Computing Practice for Large-Scale Science & Engineering Applications, Jha et al, CCPE (in press) [1] ● Critical Perspectives on Large-Scale Distributed Applications and Production Grids (Best Paper Award) [2]
Actors	User: end-scientists (e.g., chemists, climate scientists, bioinformatics), but also Science Gateways and portals. (Different resource providers: e.g., XSEDE, campuses, OSG, PRACE, EGI. There are security, policy, and accounting issues that must be addressed in addition to the issues around simply running a job.)
Prerequisites & Assumptions	User has allocations on multiple systems, and fungible allocation on XSEDE and possible other distributed cyberinfrastructures (DCIs) such as OSG and PRACE.
Steps	See UML Actor Diagram
Variations (optional)	The user may want to use not only existing XSEDE resources but also other DCI, like OSG, EGI, PRACE, or Cloud resources, viz. interoperation of XSEDE

1 https://raw.github.com/saga-project/radical.wp/master/publications/pdf/dpa-surveypaper_draft.pdf
1
2 https://raw.github.com/saga-project/radical.wp/master/publications/pdf/dpagrid2009_draft.pdf

	Resources and Federation with non-XSEDE Resources. *An interoperable Pilot-Job that would support multiple usage modes (high-throughput, high-performance as well as mixed-mode multi-component simulations) would be very useful.*
Quality Attributes	Overall time-to-completion (TTC) is one critical component - the lower the better. Other quality metrics could be the number of tasks completed. For example, there are many users with O(1000) tasks, each producing / consuming O(1GB) per task.
Non-functional (optional)	IDEAS: Interoperability, Dynamic (Resource Management), Extensibility, Adaptive and Simple.
Issues	List of issues that remain to be resolved

C Glossary

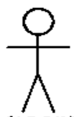
Federation: The aggregation of resources via common policies in allocation, accounting, authentication and identity management. Resources within a given "domain" are generally considered federated. Resources between different domains are federated using different models.

Interoperation: The ability to utilize distinct heterogeneous resources for a common application or user-defined goal.

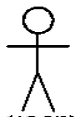
It is possible to have Federated resources that are not interoperable, but all interoperable resources are federated – at some level. Resources within XSEDE are deemed to be already federated, but are not *a priori* interoperable.

D Federation and Interoperation Use Cases

Use Case Diagram: A graphical representation of the use case.

UC 1 - Application workbad size requires as many resource as possible (AS-IS)			
Users	Actions	Objectives	Systems
 <user>	Activate credentials for all systems that user has access to	Allow job submission / file transfer to remote systems	XSEDE myproxy server, X.509 proxies, SSH keys, etc
	Deploy application 'kernel' and data on resources	Set-up job execution environment	(federated) shared filesystems, scp, gridftp, Globus-Online, iRODS
	Create batch-scripts for each resource	Map application workbad to resources	LSF, PBS, TORQUE, ...
	Run and monitor job execution across all resources	Make sure application workbad executes without errors	queuing system cmd-line tools (e.g., qstat)

AS IS: Currently the user is unable to use multiple resources on XSEDE in an uniform fashion. Where multiple resources are used, different access modes/mechanisms are employed or significant effort/laborious pre-arrangement is required. This is neither scalable nor simple to manage. Furthermore, resources do not have direct support for different usage modes (everything is a single uniform batch queue system to the user, with no distinguishing ability), nor is the user able to execute on XSEDE resources in conjunction with other resources (such as OSG, EGI).

UC 1 - Application workbad size requires as many resource as possible (TO-BE)				
	Users	Actions	Objectives	Systems
	 <user>	Activate credentials for all systems that user has access to	Allow access to remote systems	XSEDE myproxy server, X.509 proxies, SSH keys, etc
		Define application workbad / tasks	Create application workbad that is independent from any specific resource mapping	Simple text files, possibly following a standard like DRMAA or similar.
		Pass credentials and workbad to federated execution framework	Pass the burden of controlling data and task placement, execution and monitoring to the 'federation platform'	Multi-modal 'Federation platform': programmatically or via command-line tool

TO BE: The user should be able to execute tasks interoperably across different XSEDE resources which support different resource utilization models, as well as be able to utilize XSEDE resources in conjunction with OSG/EGI etc, without laborious manual pre-arrangement.

Much has been written about Interoperation. It can be provided at multiple-levels and along different application "vectors" (development, deployment and execution stages).

E Appendix

This use case is derived (distilled) from multiple distinct application usage scenarios. We list two examples here, but there are multiple NSF funded projects that can be distilled into the above discussed use-case.

1. Data-Intensive (Bioinformatics) Workflows: (e.g, BWA, Bowtie, BFAST): There are an increasingly large number of data-intensive application workflows (our experience is coincidentally focused around bioinformatics/next-generation sequencing based applications), that require distribution due to a plethora of reasons. Some are due to the fact that the data is fundamentally distributed but too large to move around efficiently. In such cases, placing the computational tasks to take advantage of existing data placement/localization is not only a performance enhancer, but

important. In other cases, distribution is important because localizing all the data-intensive computing onto one machine results in I/O sub-system saturation.

A standards-based interoperation for such data-intensive workflows using advances in data-cyberinfrastructure, such as iRODS, GFFS and SRM are required for flexible, dynamic and scalable execution. However, too many distinct, incompatible and point solutions exist.

2. High Performance High Throughput (HPHT): (Current XSEDE XRAC PIs: Bishop, Levy, Coveney). There are many molecular simulations that require multiple instances of the same kernel, either to implement algorithms that provide enhanced sampling or better statistics. Most examples involve zero coupling between the kernels (other than possible data dependencies), but some have weak dependencies between the kernels at runtime. Multiple XSEDE PIs/users need this increasingly important mode.